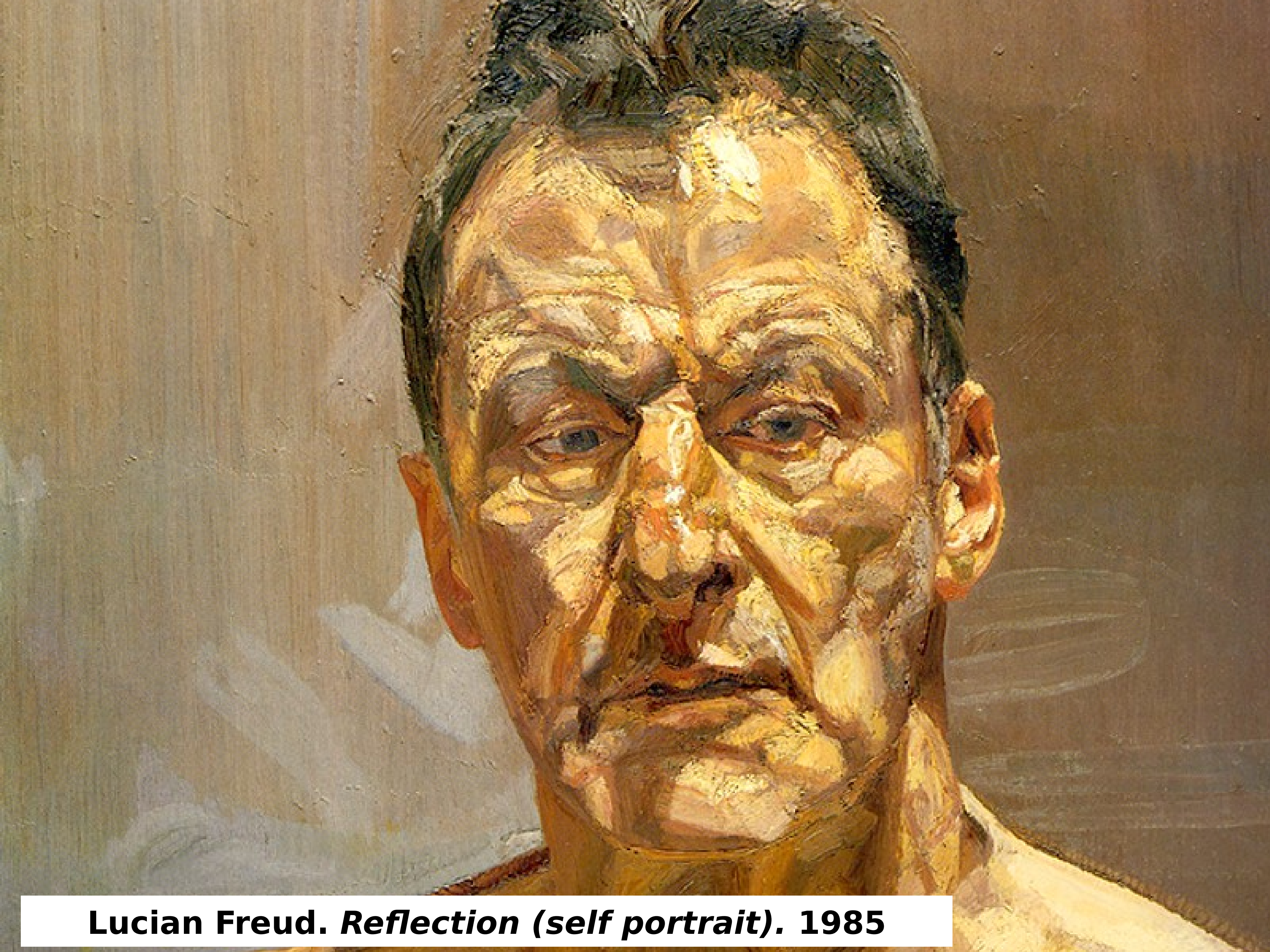
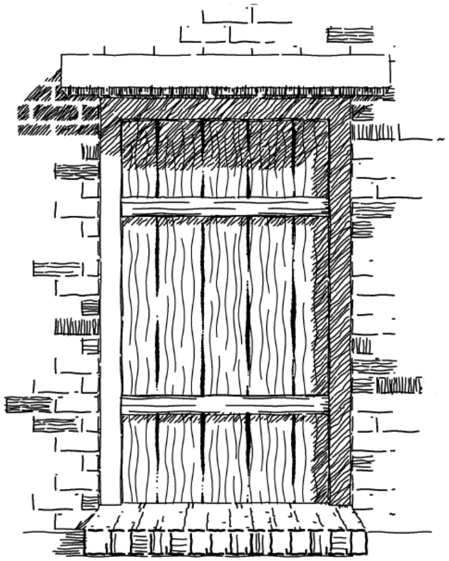
Stroke-Based Rendering

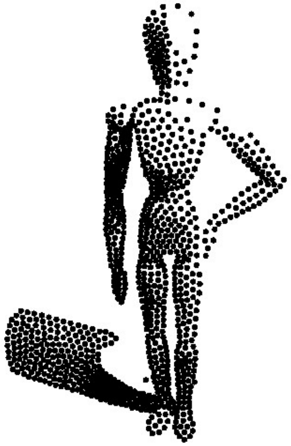
Aaron Hertzmann
University of Washington

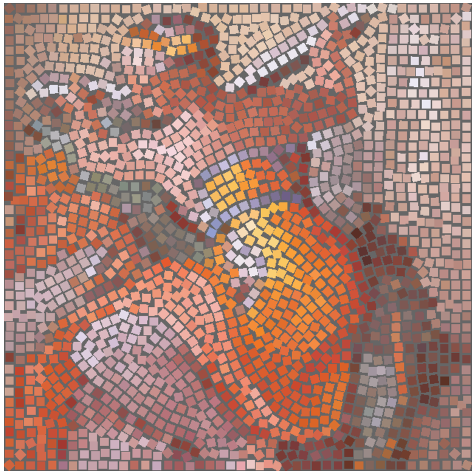


Lucian Freud. ***Reflection (self portrait).*** **1985**

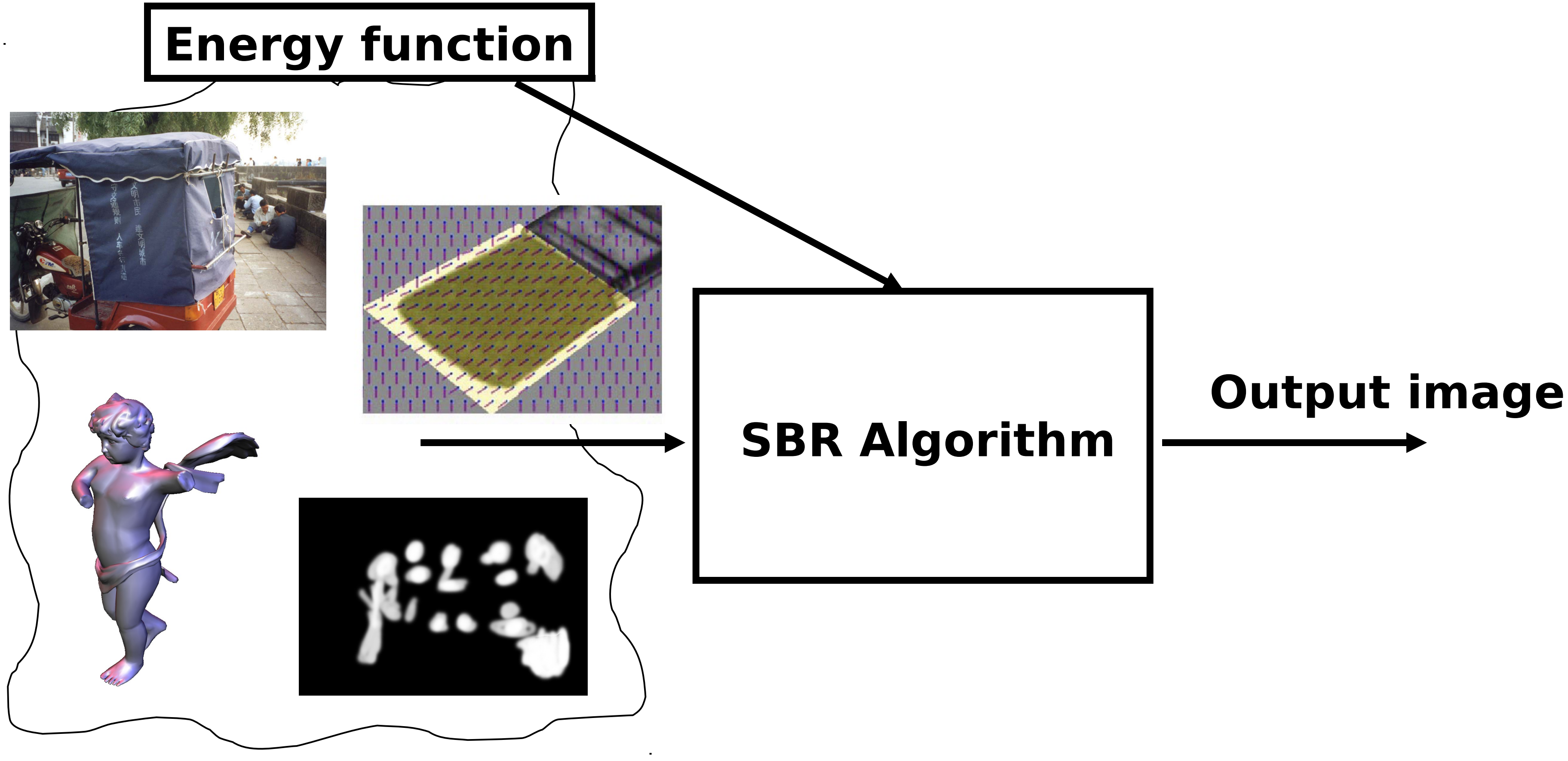
Many SBR algorithms…







Unified view



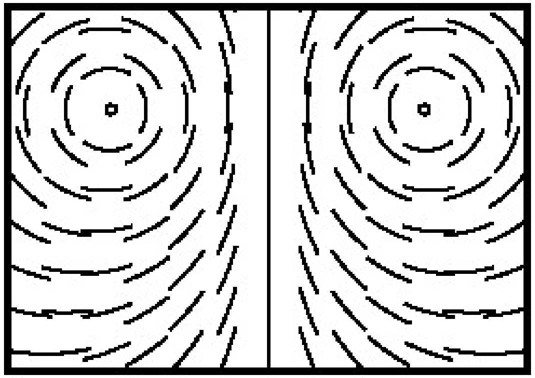
Energy function

SBR Algorithm

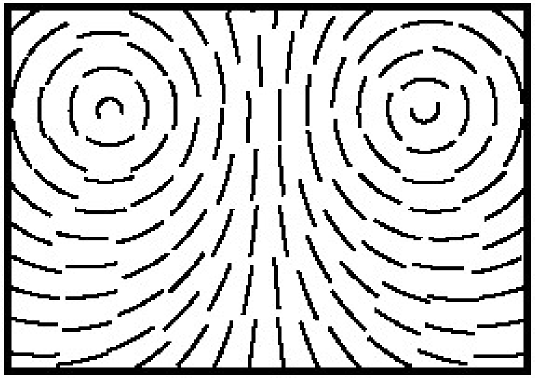
Output image

Input Specifications

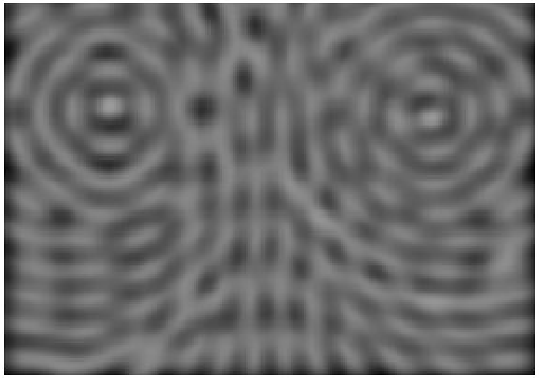
Vector field illustration



Input



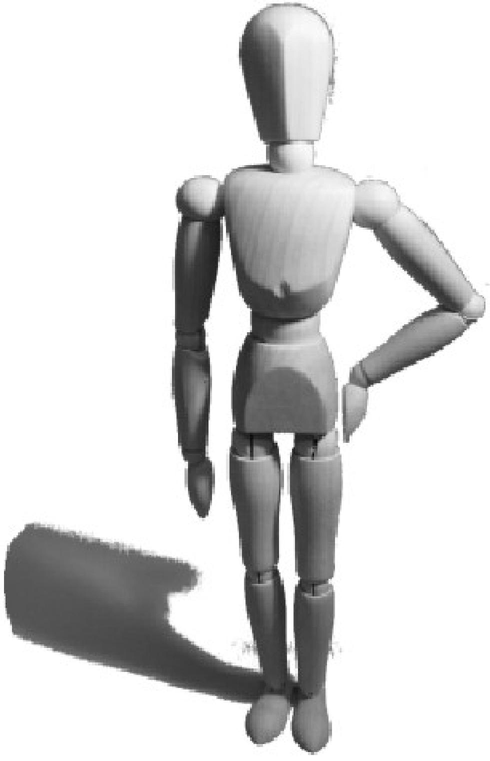
Output



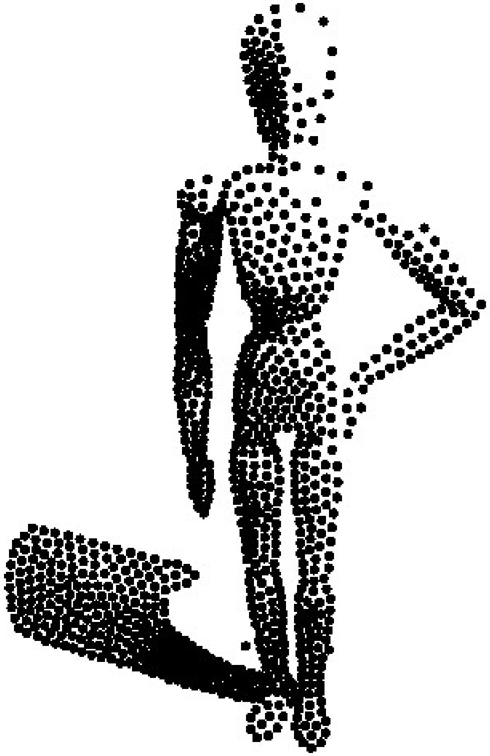
Energy function: $$\sum \left\| G - B(x, y) \right\|^2$$

Turk and Banks, SIGGRAPH 96

Stippling



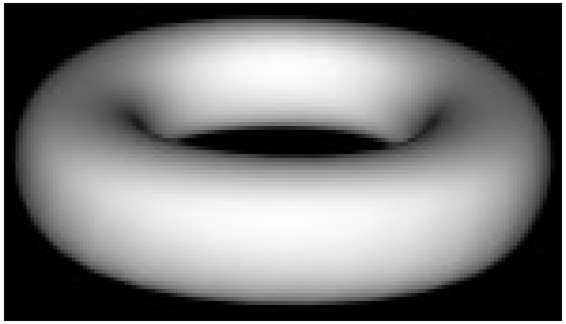
Input



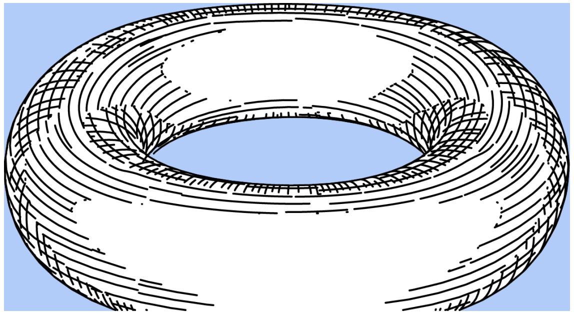
Output

Secord, NPAR 02

3D illustration



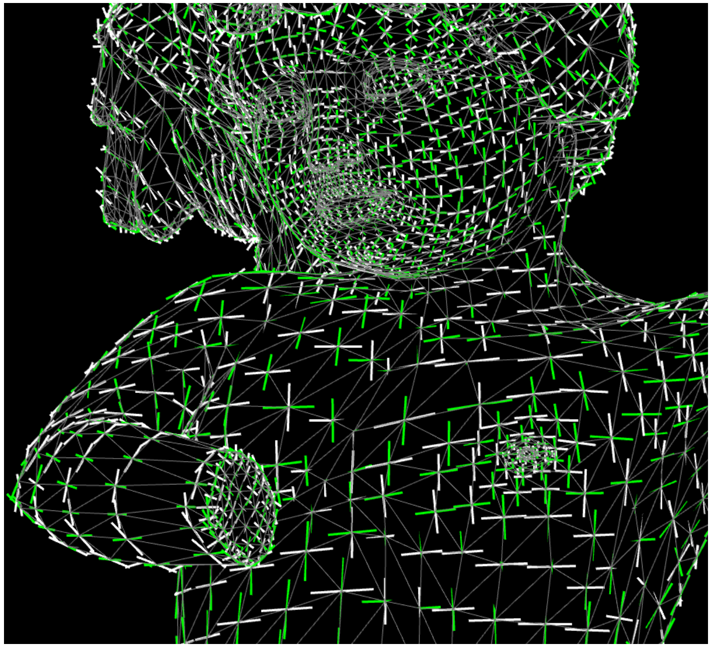
3D model/ intensity



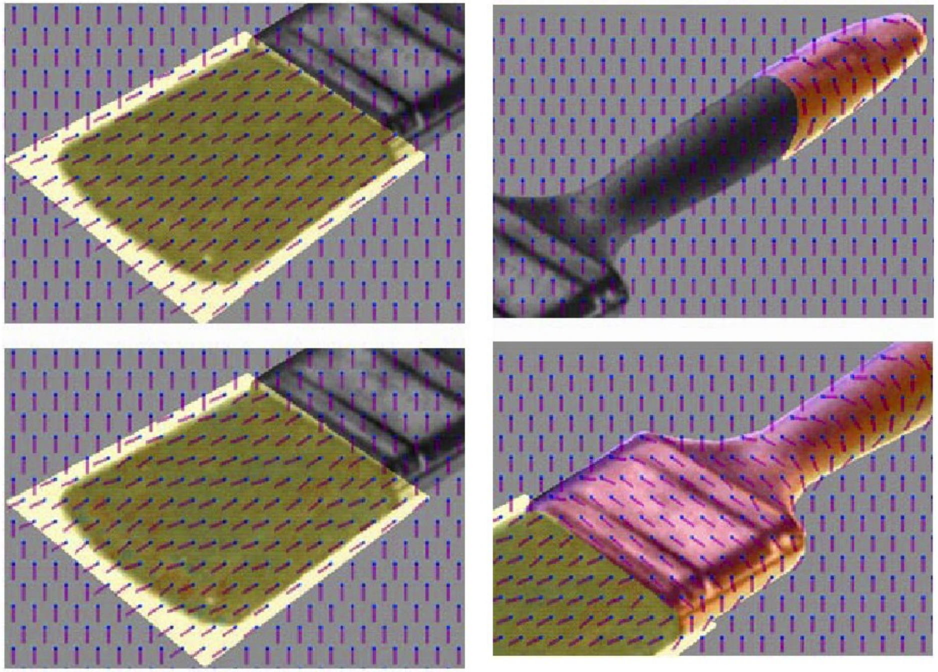
Hatching

Hertzmann and Zorin, SIGGRAPH 00

Hatching orientations

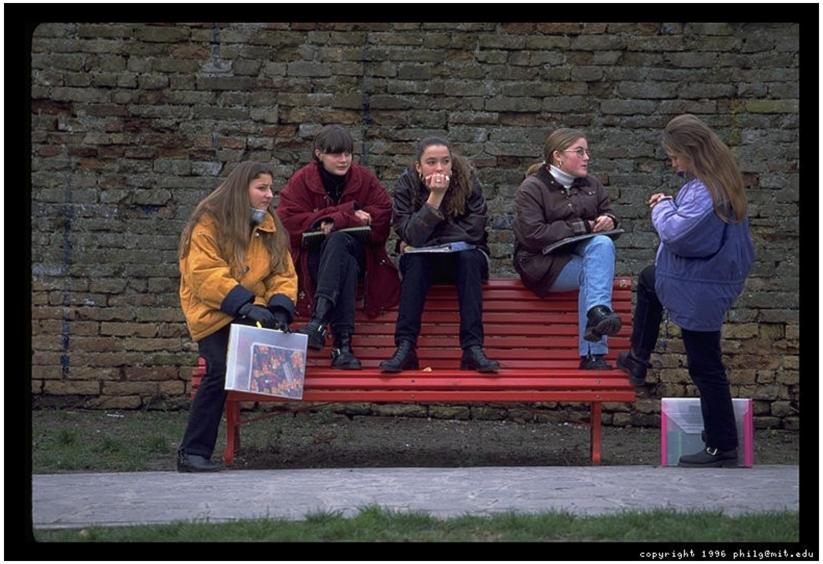


From 3D data

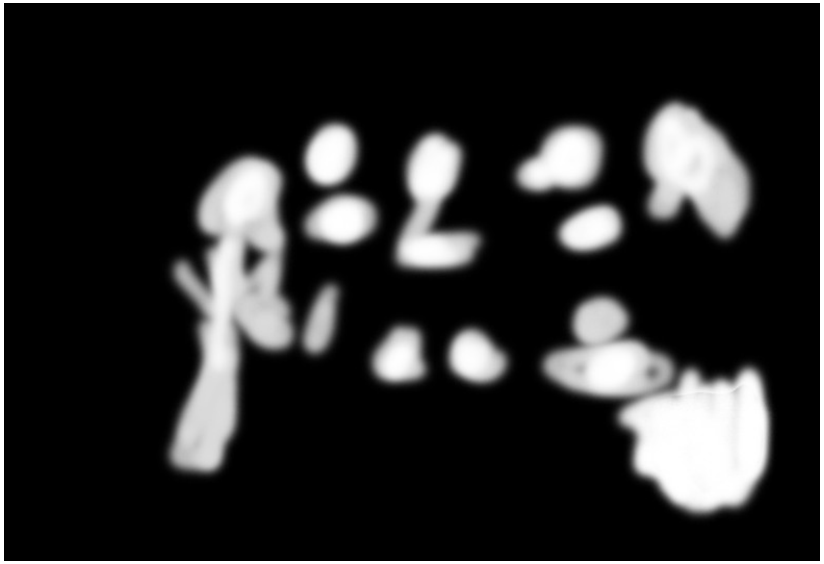


User input

Emphasis variation

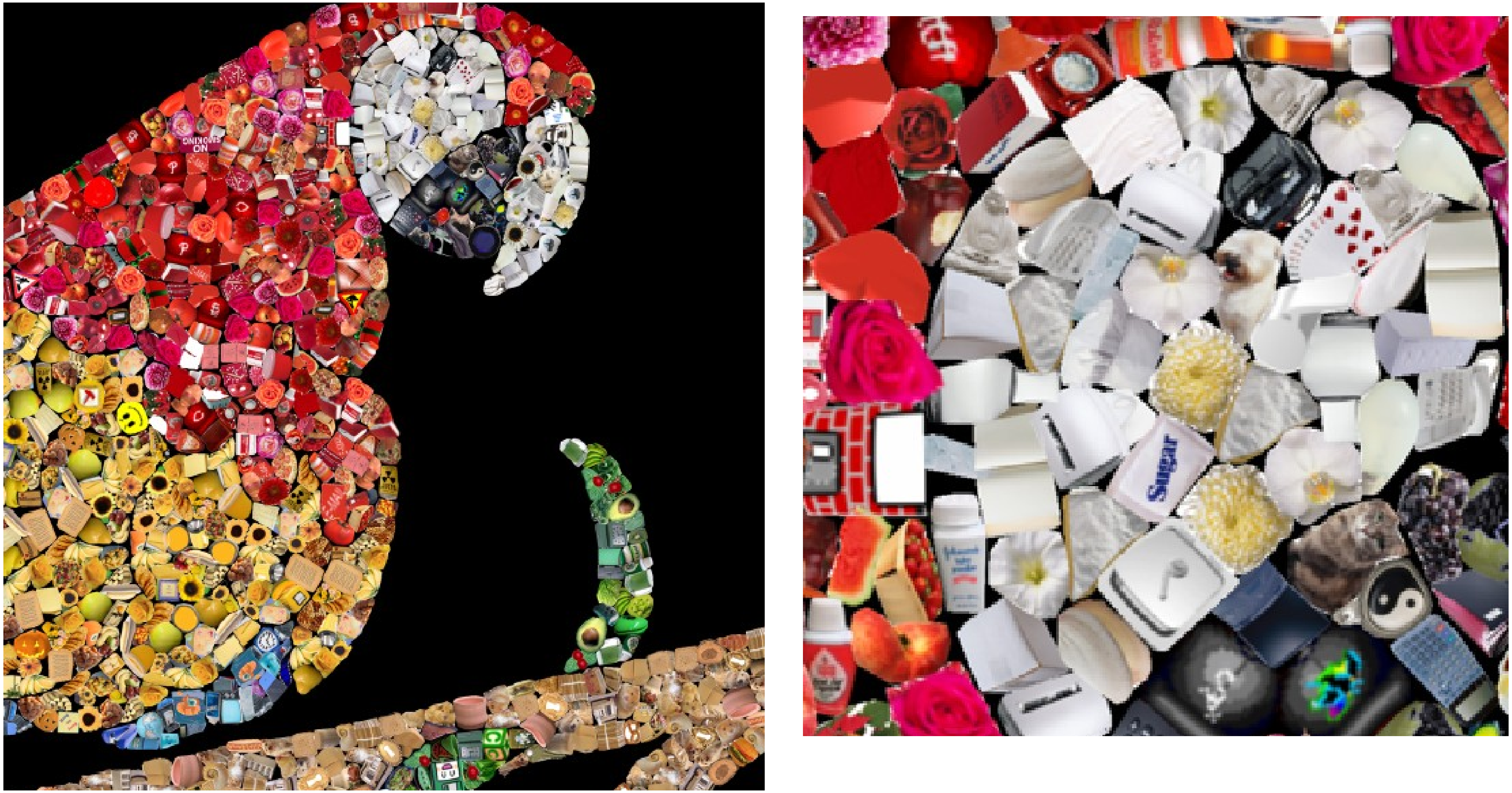


Source image



Weights

Jigsaw image mosaics



Kim and Pellacini, SIGGRAPH 02 (Friday morning)

Focus on automation

- Painting software is an artistic tool
- Computer performs repetitive tasks
  - e.g. placing brush strokes
- This talk: “High-level” paintbox

Outline

- Greedy algorithms
- Optimization algorithms

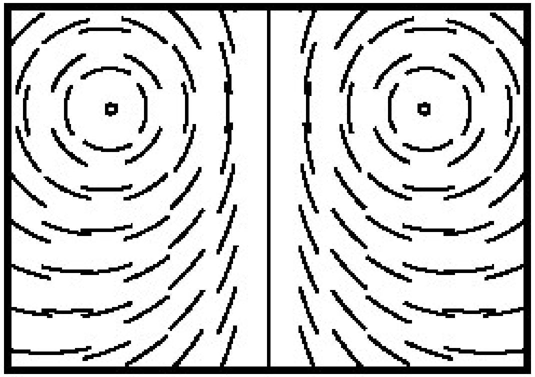
Greedy algorithms

Greedy algorithms

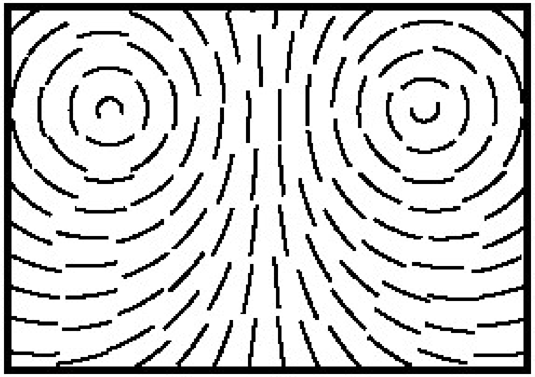
```
while not done
    pick a starting point
    create a stroke
```

Vector field visualization

- Problem statement



Input

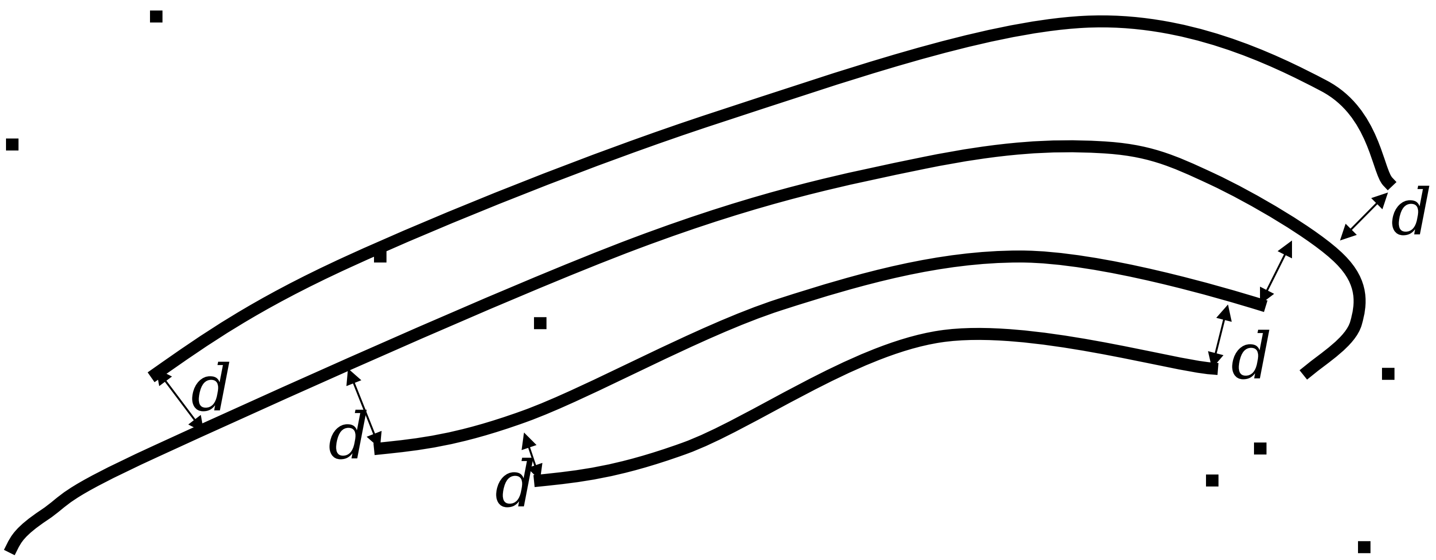


Output

Turk and Banks, SIGGRAPH 96

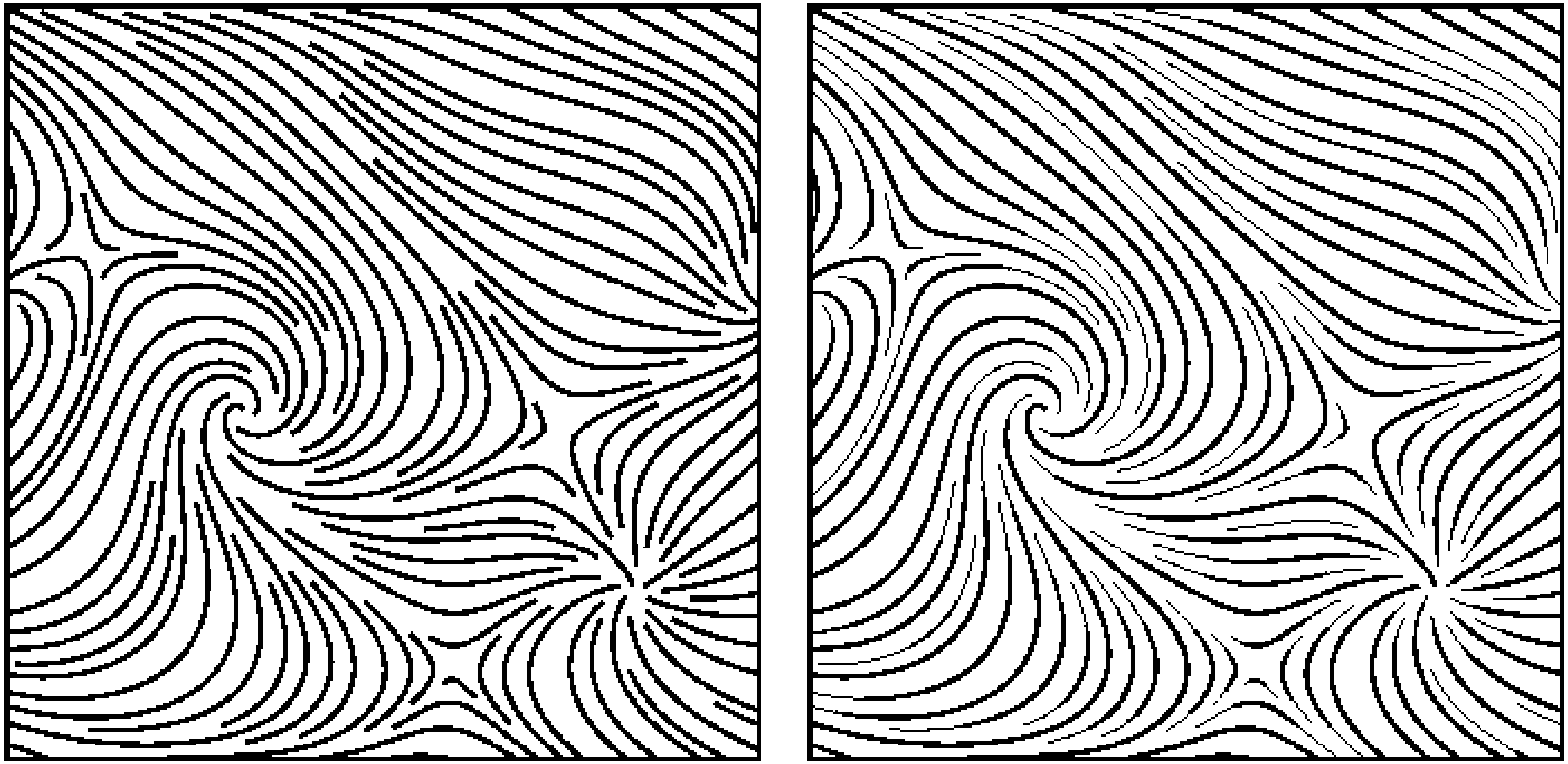
Greedy algorithm

Goal: spacing *d* between strokes



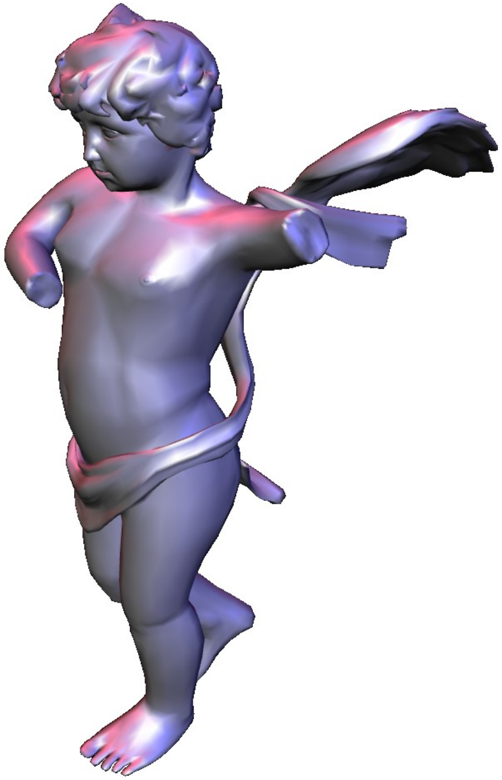
Jobard and Lefer, EVSC 97

Vector field visualization



Jobard and Lefer, EVSC 97

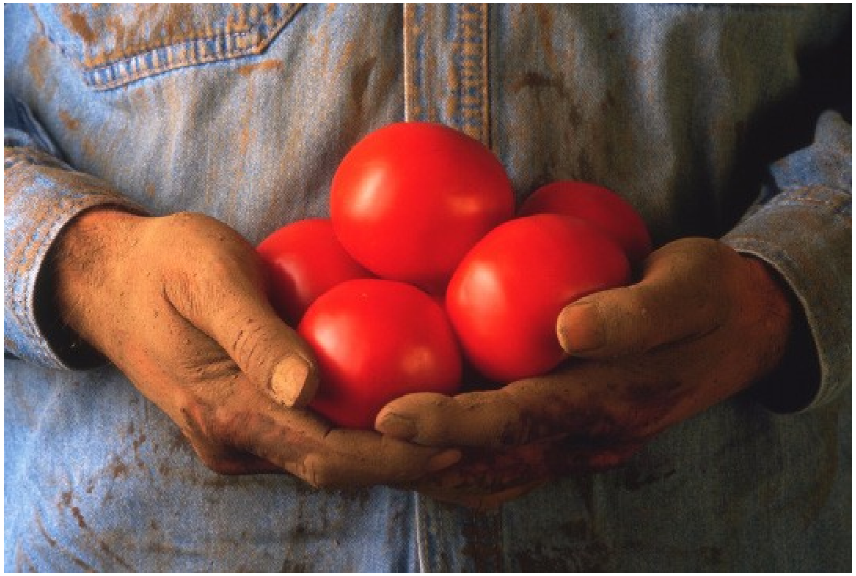
Illustrating smooth surfaces



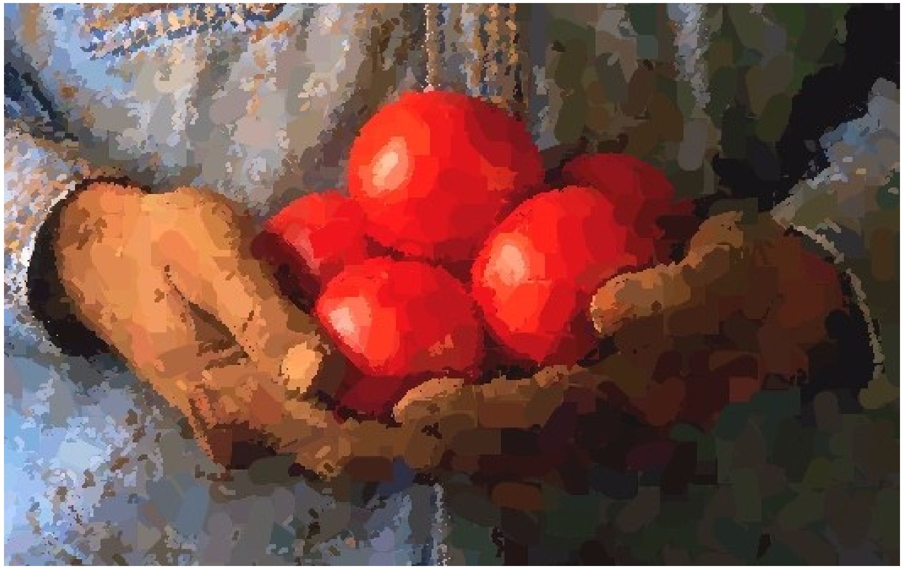
Hertzmann and Zorin, SIGGRAPH 00

Painterly rendering

Problem statement

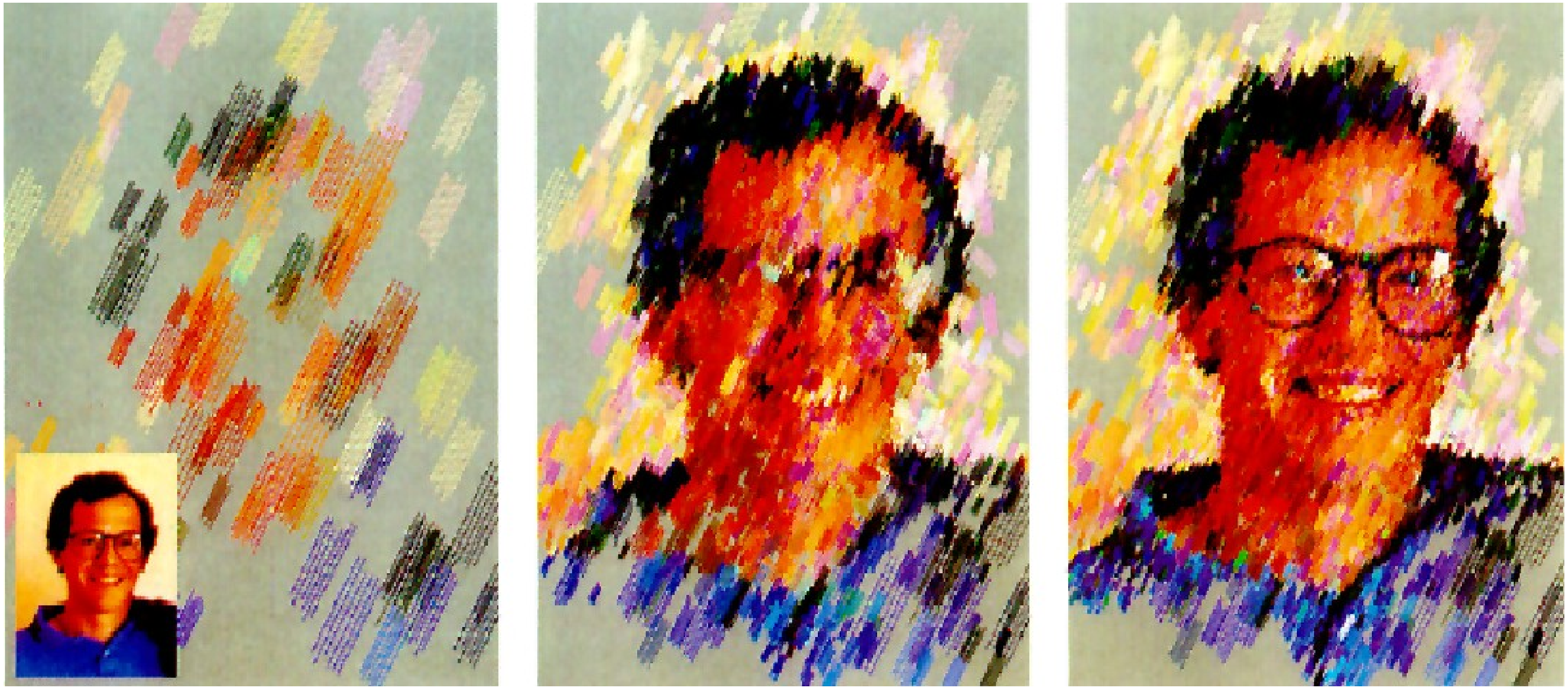


Input image



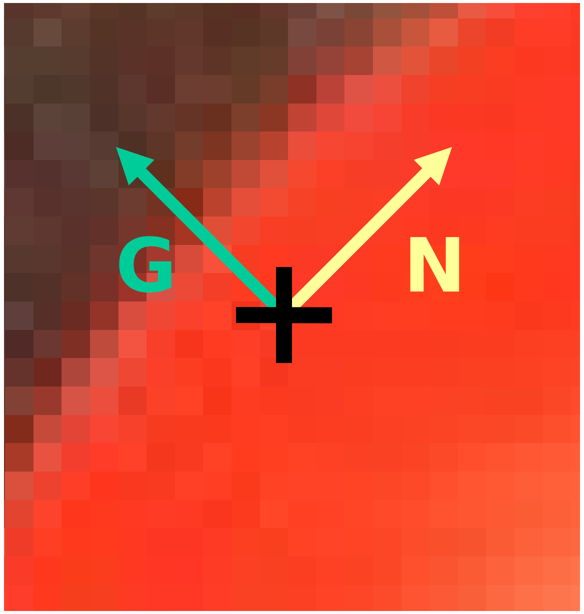
Painting

The Impressionist

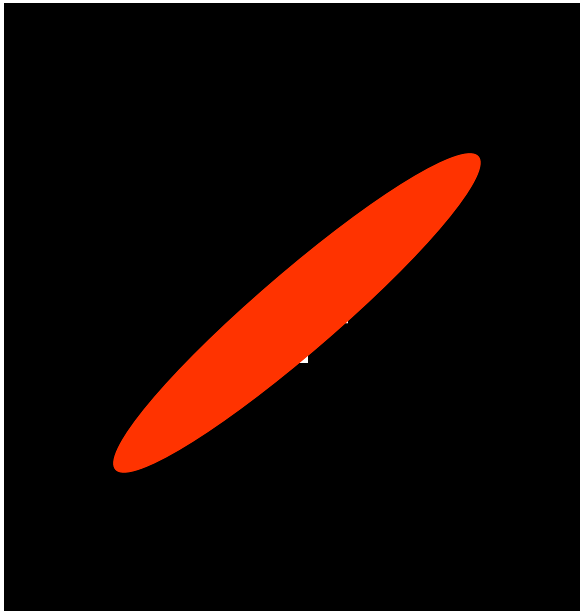


Haeberli, SIGGRAPH 90

Stroke orientations

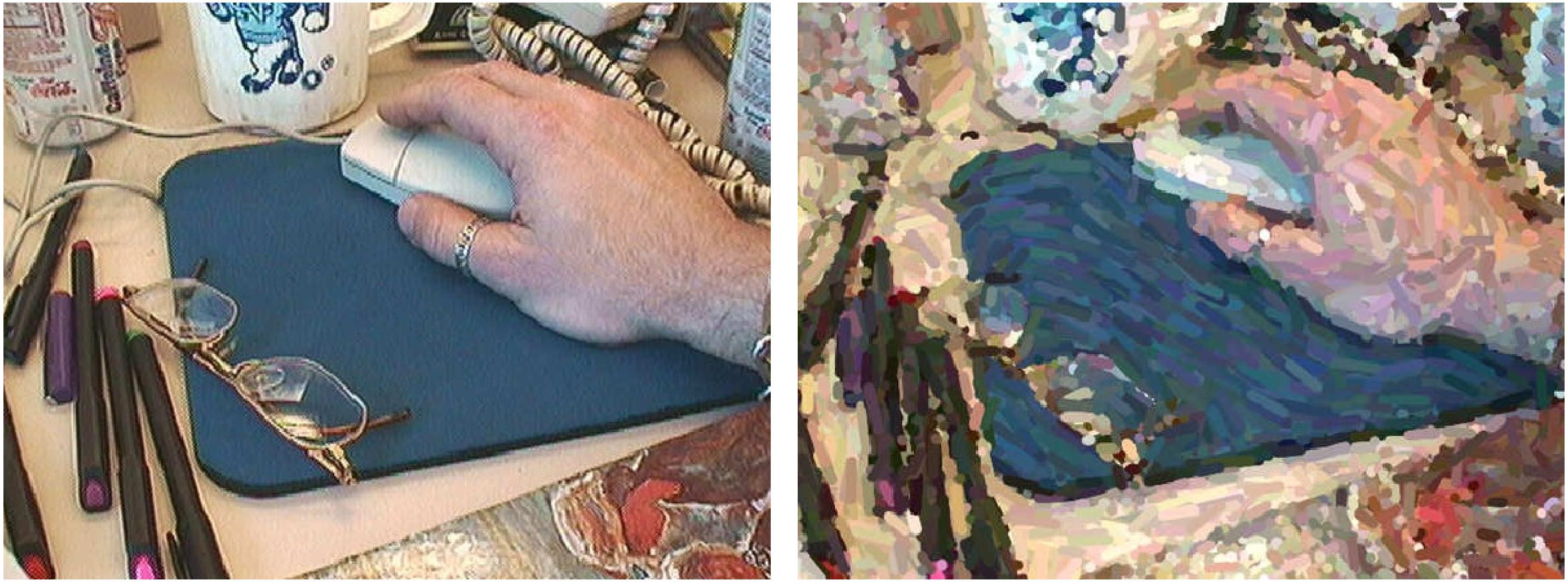


Source Image



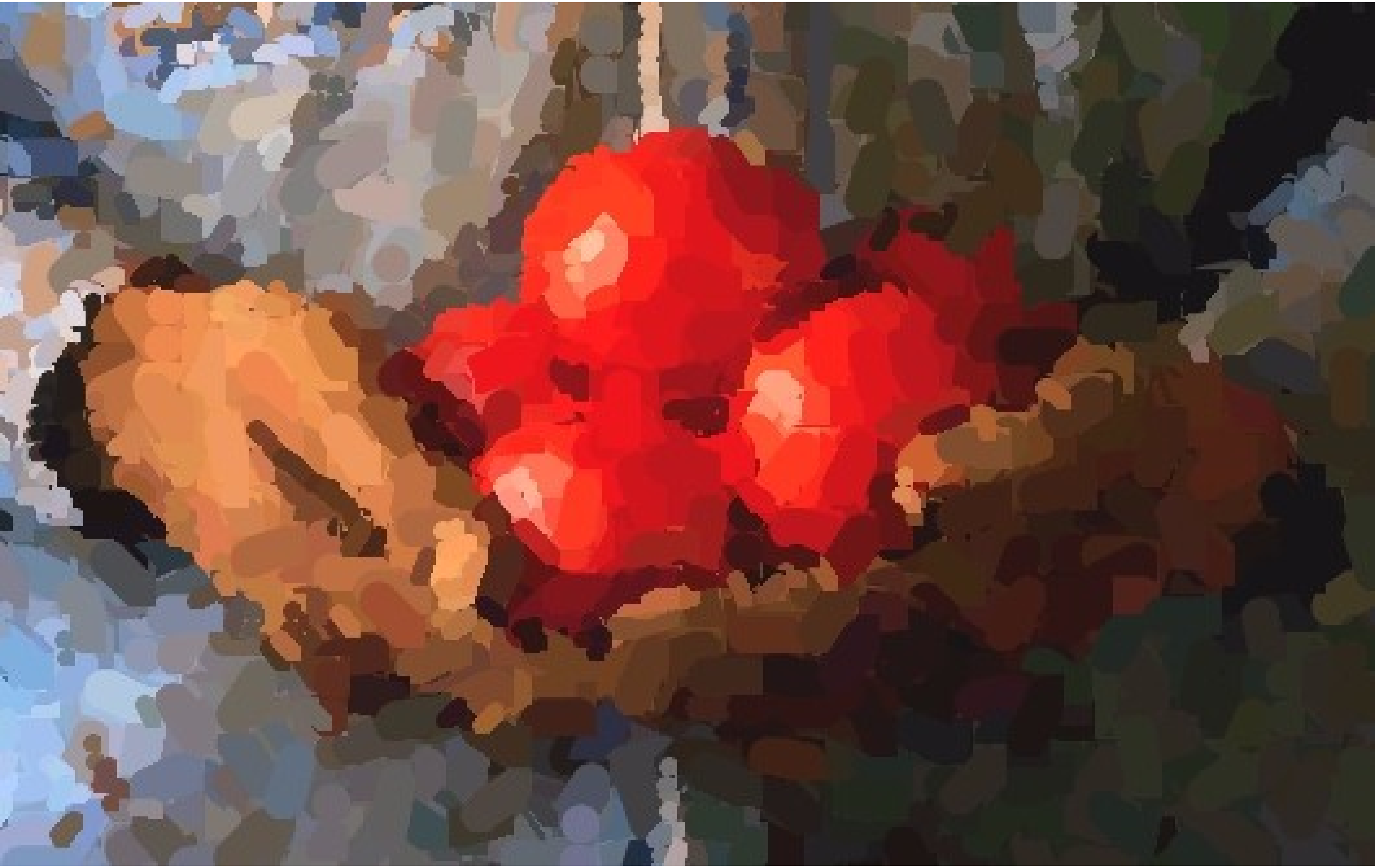
Painting

Gallery effects

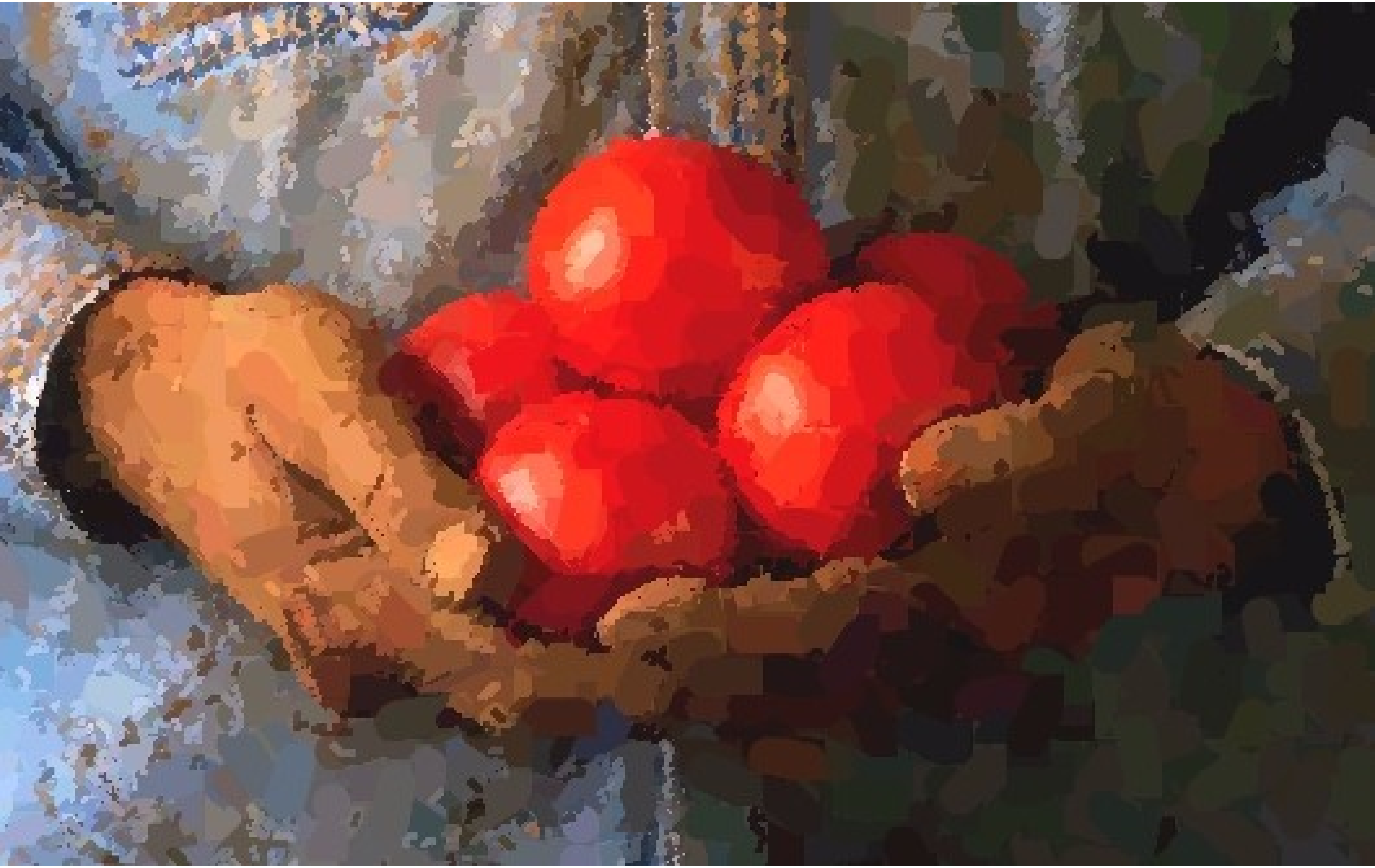


Litwinowicz, SIGGRAPH 97

First Layer

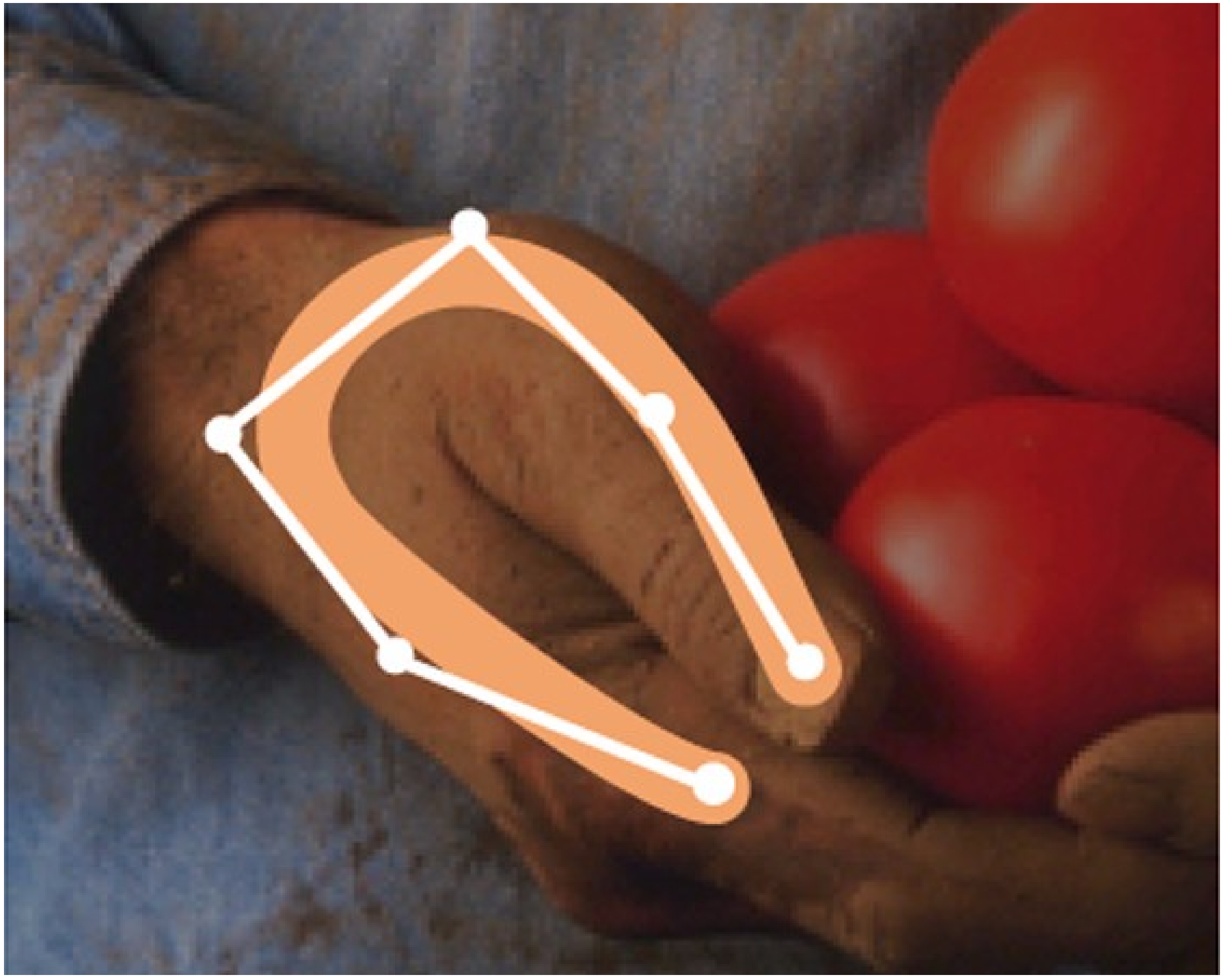


Second Layer

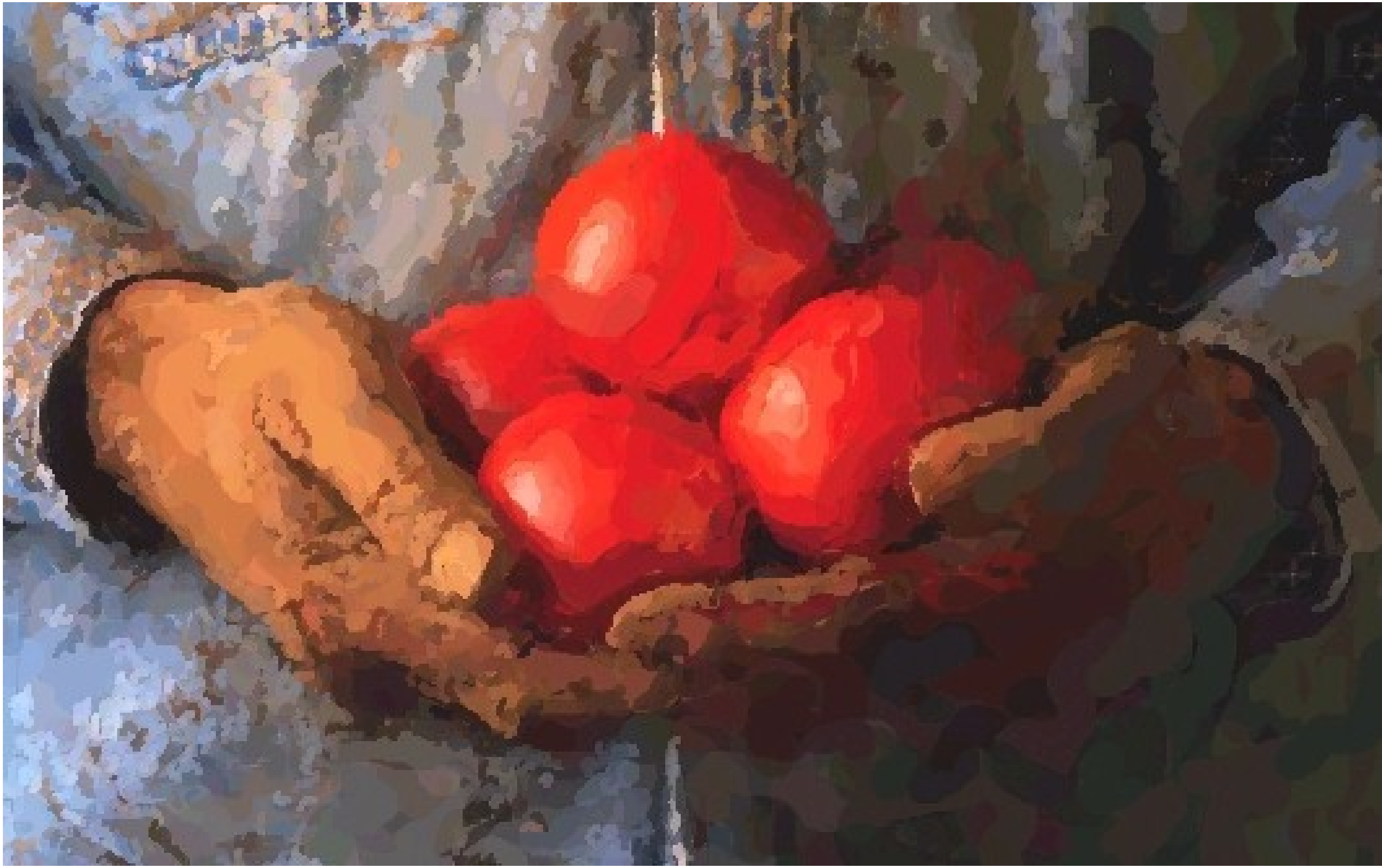


Final Painting

Brush strokes



Curved strokes



“Impressionist” tomatoes

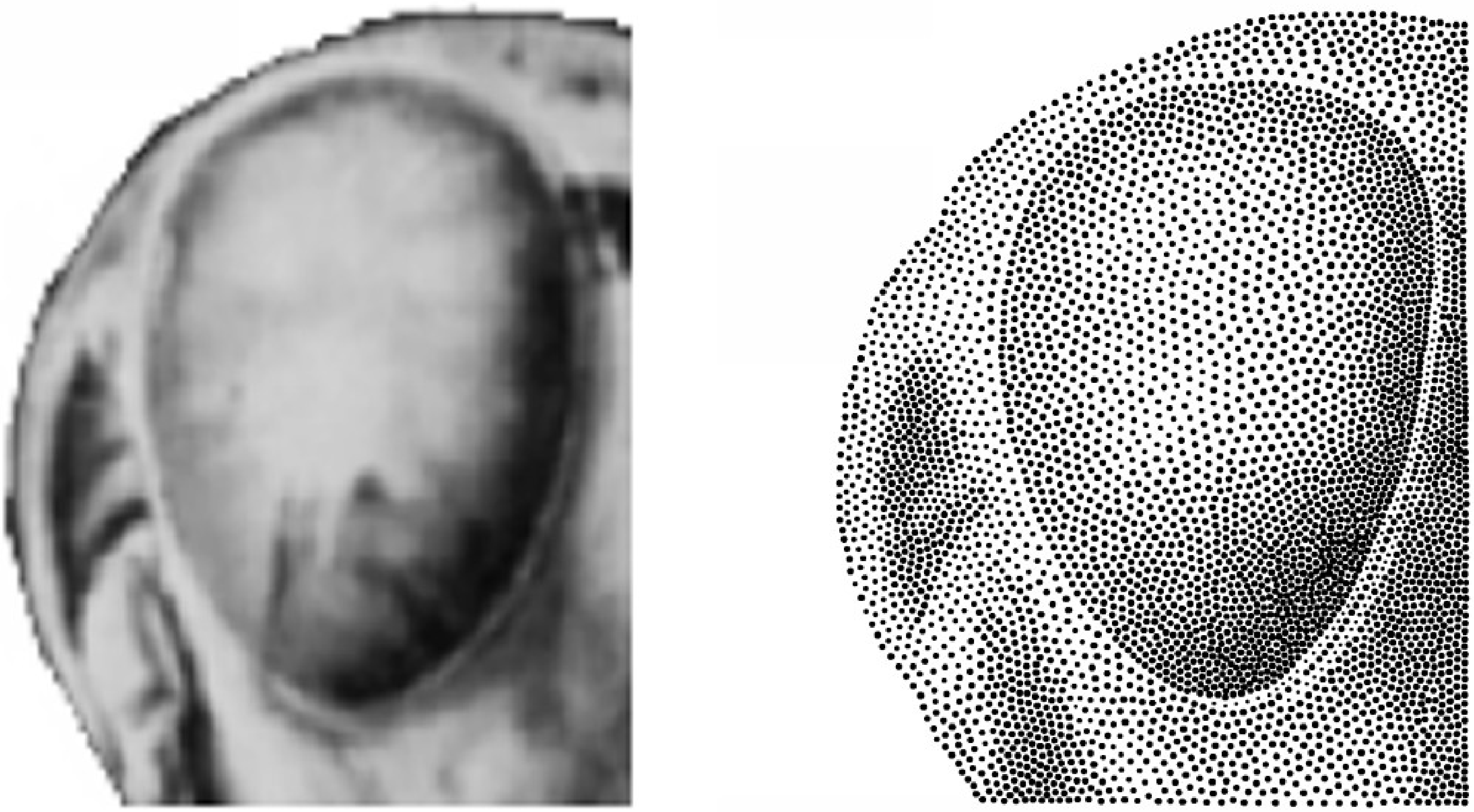
文明市民 建文明城市
守交通规则 人车各行其道
21
黄山市

Optimization algorithms

Optimization algorithms

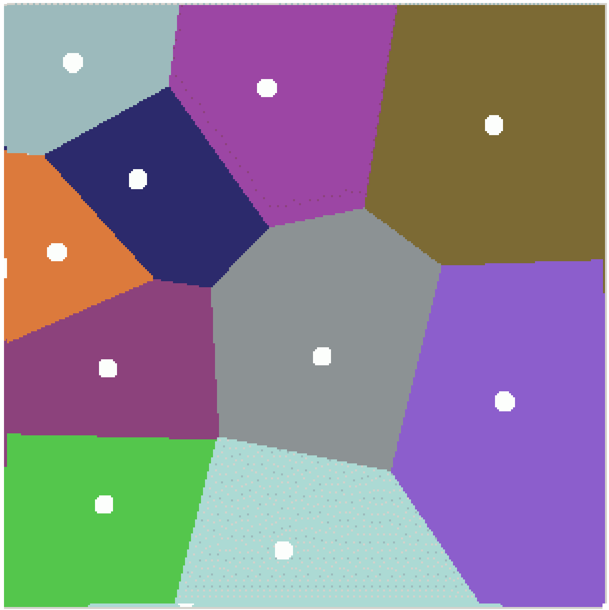
- Define formal energy and constraints
- Iteratively improve the rendering

Stippling



Deussen et al., EG00

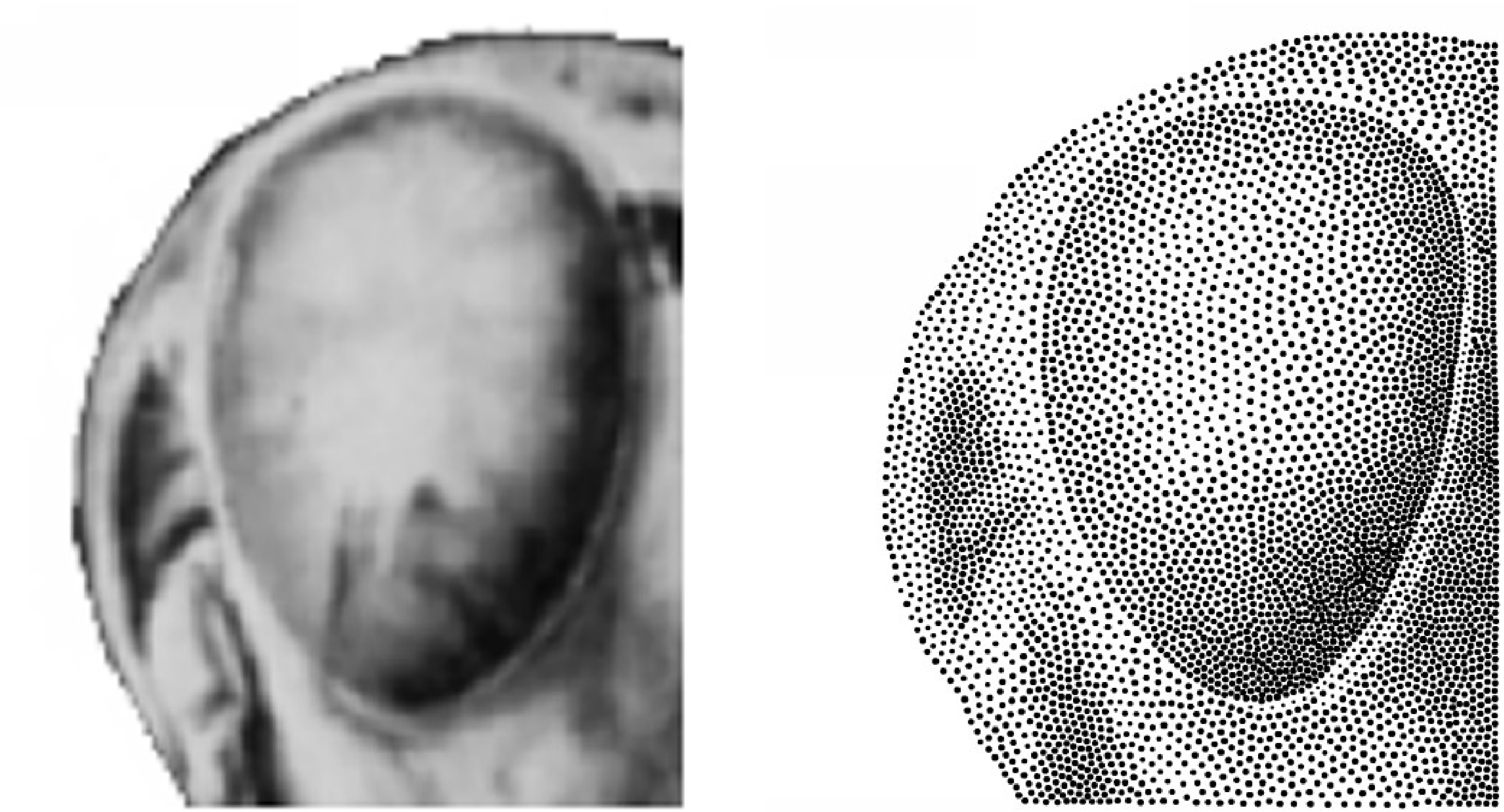
Lloyd’s algorithm



With graphics support: Hoff, SIGGRAPH 99

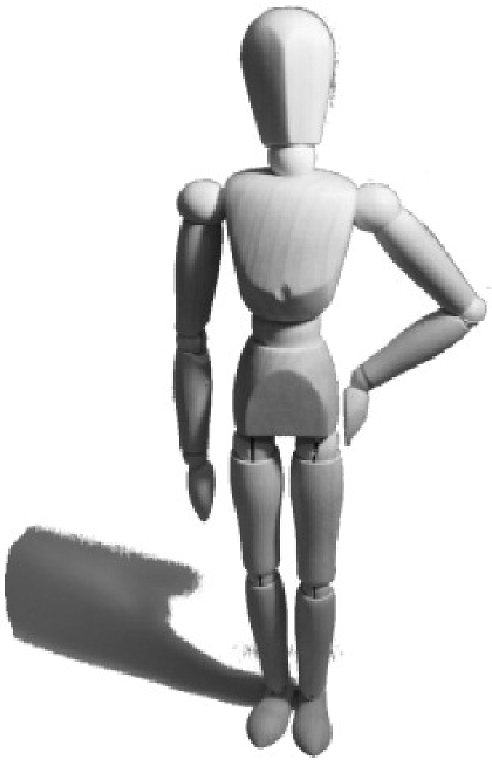
Hausner, SIGGRAPH 01

Stippling

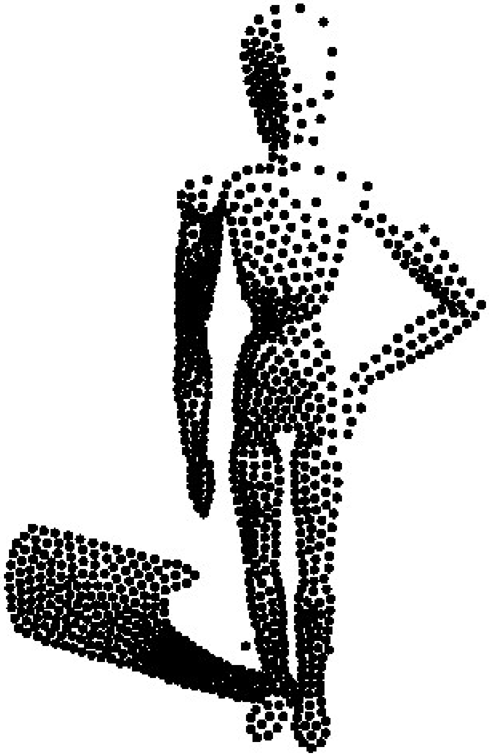


Deussen et al., EG00

Weighted stippling



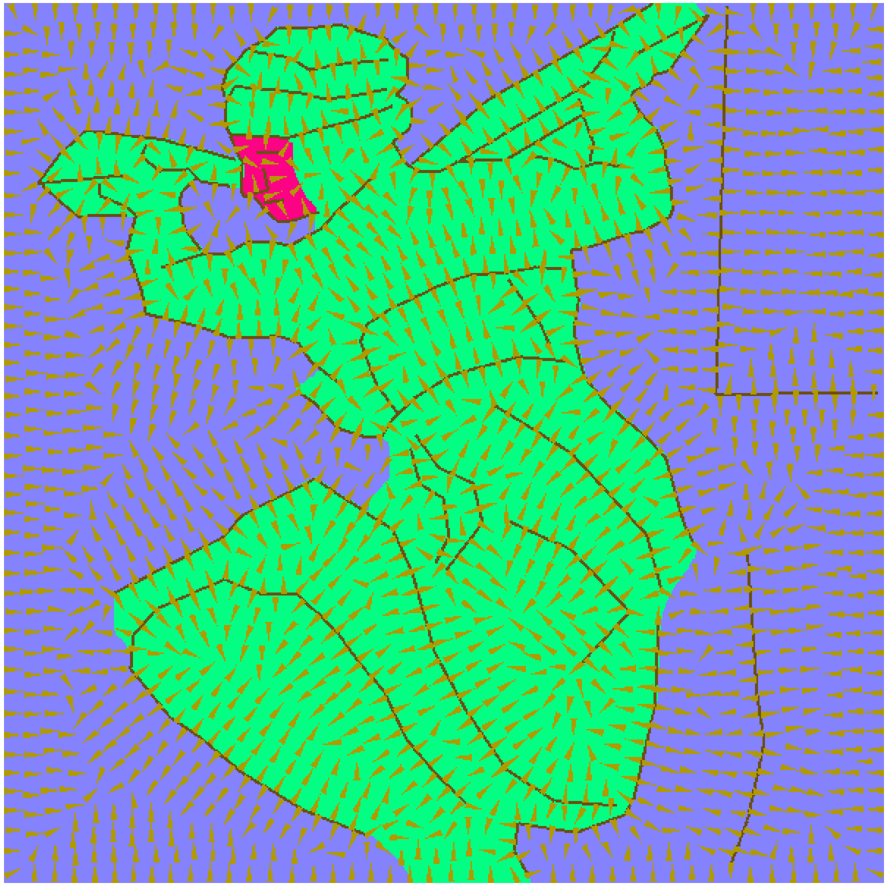
Input



Output

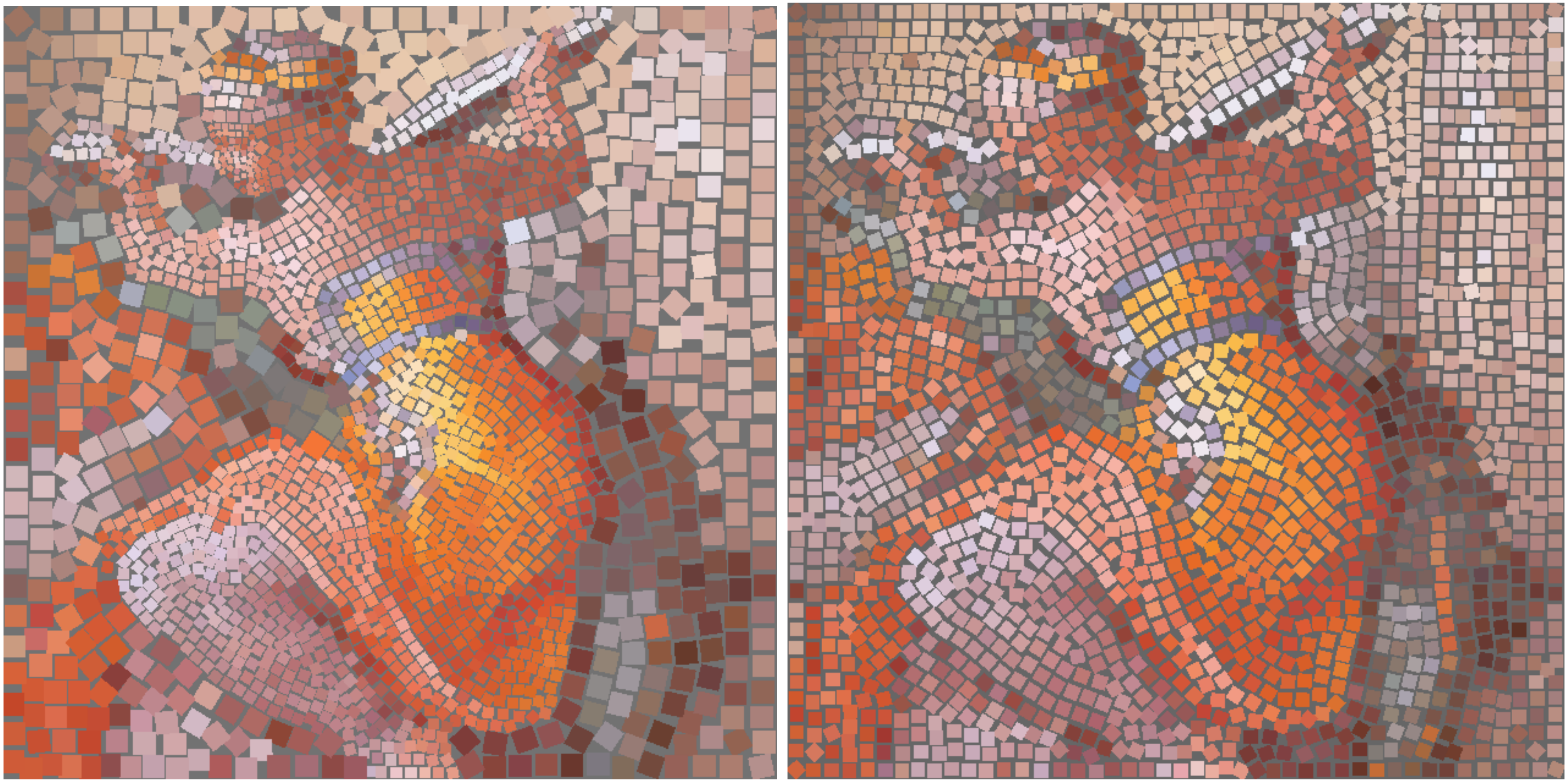
Secord, NPAR 02

Tile mosaics



Hausner, SIGGRAPH 01

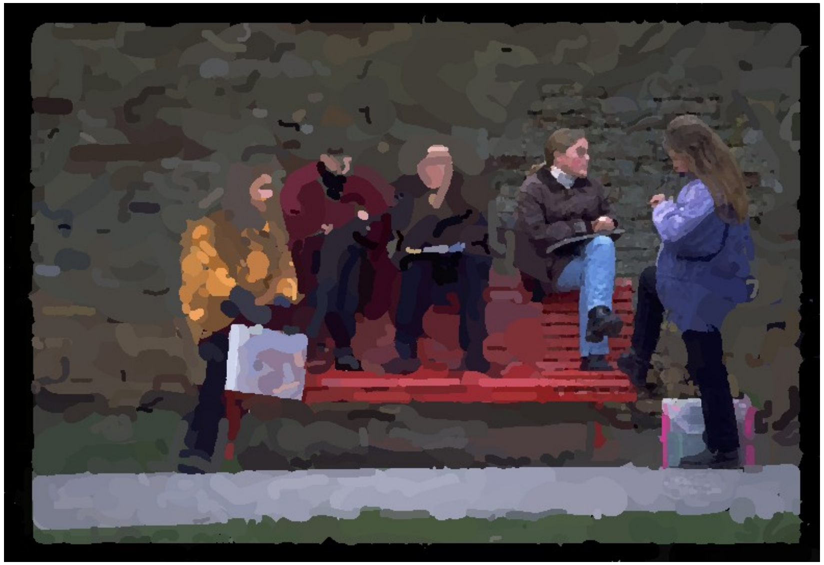
Tile mosaics



Hausner, SIGGRAPH 01

Paint by relaxation

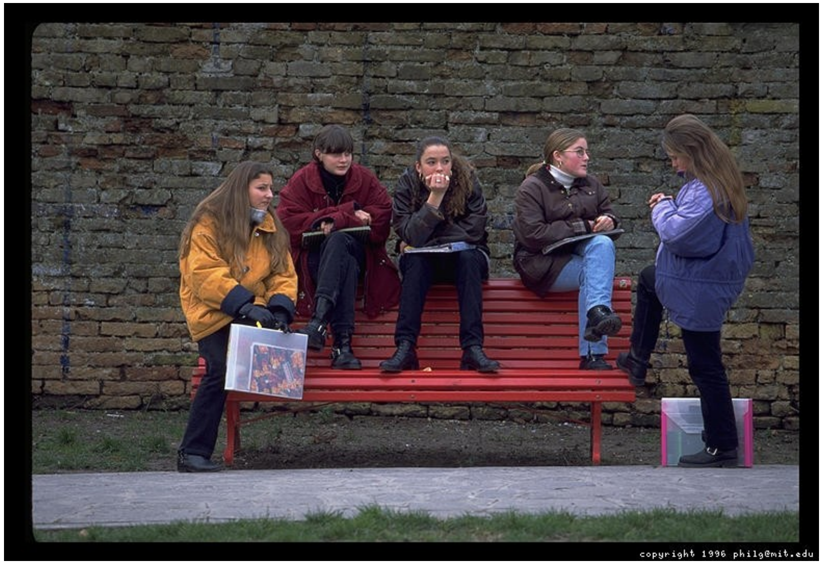
Source image



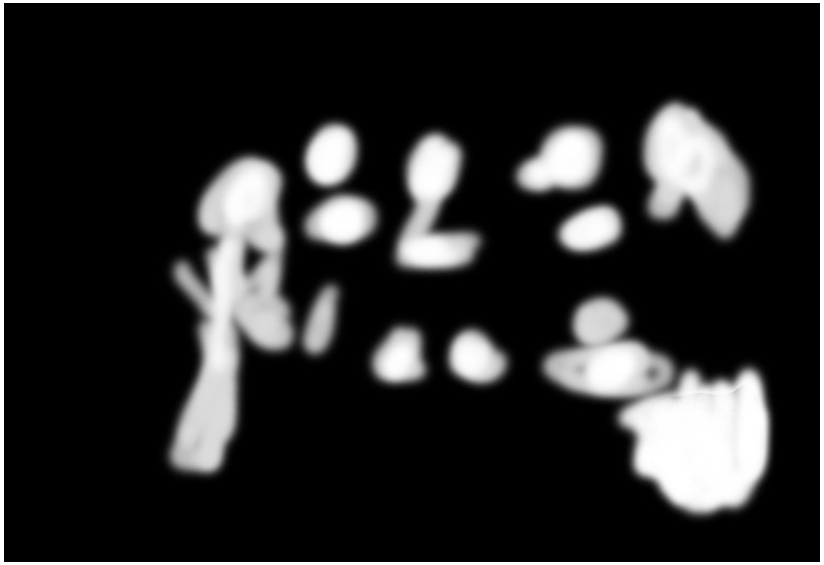
Painting

$$\sum \left\| I(x, y) - P(x, y) \right\|^2 + N$$

Paint by relaxation



Source image



Weights

$$\sum w(x, y)\left\| I(x, y) - P(x, y) \right\|^2 + N$$

Summary

- Stroke-based rendering:
  - *Place strokes to minimize an error function*
- Two algorithmic approaches:
  - **Greedy algorithms**
    - *more intuitive to design*
    - *harder to interpret and control*
  - **Optimization algorithms**
    - *formal specification*
    - *slower*